CLAIMS

What is claimed is:

1. A method for routing a multi-layered integrated circuit, comprising:

receiving parameters for an integrated circuit having n layers, wherein n is at least two;

constructing a routing graph for a level k of the n layers of the integrated circuit, the k level partitioned into k–level tiles, wherein at least one edge is provided to join a first tile and a second tile in the k level routing graph;

calculating capacities of the at least one edge joining the first tile and the second tile, the edge at least one of regular and normal;

adding occupancies based on previous level net routing $(k + 1)$ and calculating penalties of edges; and

routing nets based on the routing graph, calculated capacities and added occupancies.

2. The method as described in claim 1, wherein after routing nets, if k is more than zero, the constructing, calculating, adding and routing steps are repeated after subtracting one from k.

3. The method as described in claim 1, wherein after routing nets, if k is equal to zero, the adding and routing steps are repeated after subtracting one from k.

4. The method as described in claim 1, wherein after routing nets, if k is less than zero, legal net routing is obtained.

5. The method as described in claim 1, wherein routing nets includes routing nets in a net routing area and routing open nets in all integrated circuit areas.

6. The method as described in claim 1, wherein the parameters include at least one of netlist, grid lines slop, distance between neighboring grid lines, and wire blockage information.

7. The method as described in claim 6, wherein the parameters include parameter information for *n* layers of the chip.

8. The method as described in claim 1, wherein a tile is a square section, with one side of each tile of the layer directed along a layer grid line.

9. The method as described in claim 8, wherein a tile is sized so that each tile side is approximately equal to one grid.

10. The method as described in claim 8, wherein grid lines are positioned generally through a middle of a tile.

11. The method as described in claim 1, wherein the routing graph includes a vertex including a tile.

12. The method as described in claim 1, wherein capacity for a regular edge is equal to the height of a tile corresponding to the edge expressed in grids minus the number of grid covered by routing blockages.

13. The method as described in claim 1, wherein capacity for a normal edge is half of the number of vias included in the corresponding tiles.

14. The method as described in claim 1, wherein adding occupancies based on

previous level net routing includes if pervious level net routing is not available, occupancies are set to zero.

15. The method as described in claim 1, wherein the penalty for passing through an edge is a function of the quotient occupancy/capacity and of the length of the edge.

16. The method as described in claim 1, wherein the penalty at least one of increases as a function of occupancy/capacity and increases as a function of length.

17. The method as described in claim 16, wherein the penalty is computed as follows $length \bullet e^{occupancy/capacity} + length$.

18. The method as described in claim 1, wherein routing nets includes arriving at a set of edges such that any two different pins of the net are connected by the set.

19. The method as described in claim 1, wherein routing nets includes implementing a procedure to route nets, wherein the input of the procedure is a net, and the output is a routing of the net.

20. The method as described in claim 19, wherein the procedure to route nets includes if an input net has two pins including a first pin and a second pin, a neighborhood of the first pin is grown until the neighborhood intersects the second pin.

21. The method as described in claim 20, wherein a least-penalty path connecting the first pin and the second pin is chosen.

22. The method as described in claim 20, wherein the procedure to route nets includes if an input net has three pins, the neighborhoods of all pins are grown until a vertex is reached in another pin.

23. The method as described in claim 22, wherein if a vertex in the intersection of all neighborhoods is provided, a sum of penalties to all three pins from that vertex.

24. The method as described in claim 23, wherein there is a vertex in the intersection of all neighborhoods, the sum of penalties of all three pins are found from that vertex such that a vertex P_0 for which that sum is minimal is found, and is denoted by Pen_0, then, for each P_i (i = 1, 2, 3), the sum of penalties to other two pins is found from the pin P_i and that sum is denoted by Pen_i, so that if $Pen_0 = min\{Pen_i, i = 0, 1, 2, 3\}$ then the net routing is the union of the three least-penalty paths from P_0 to all pins P_1, P_2 and P_3, if $Pen_1 = = min\{Pen_i, i = 0, 1, 2, 3\}$ then the net routing is the union of the two paths from P_1 to pins P_2 and P_3.

25. The method as described in claim 20, wherein the procedure to route nets includes more than three pins, the net is partitioned into at least two subnets.

26. The method as described in claim 25, wherein for each pin, a center of gravity is calculated and then the pins are sorted in ascending order of abscissae of the center of gravity, such as let $\{x_1, ..., x_n\}$ be the ordering set of abscissae, x_m is found such that $x_{m+1} - x_m = max\{x_{i+1} - x_i, i = 1, 2, ..., n - 1\}$, the point x_m divides net into 2 subnets wherein the first subnet includes from m first pins and the second subnet contains another pins.

27. The method as described in claim 26, wherein the net $\{P_1, \ldots, P_n\}$ is portioned into 2 subnets $\{P_1, \ldots, P_m\}$and $\{ P_{m+1}, \ldots, P_n\}$, the neighborhoods of the following set:

$$\bigcup_{i=1}^{m} P_i$$

and may be grown until a vertex from the following set is reached:

$$\bigcup_{i=m+1}^{n} P_i$$

the least-penalty path from one set to another is chosen and this path is included in the routing of the net, after which, a Procedure to Route Net is recursively applied to the subnets $\{P_1, \ldots, P_m\}$and $\{ P_{m+1}, \ldots, P_n\}$, the procedure based on the number of pins included in the subnets.

28. The method as described in claim 1, wherein routing nets includes implementing a procedure to grow neighborhoods.

29. The method as described in claim 28, wherein the input of the procedure to grow neighborhoods is a set *Sour* of vertices of the routing graph, which may be referred to as a source set, and a set $Dest \subseteq V$, referred to as a destination set, the output of Procedure to Grow neighborhoods includes a number $d = Dist(Sour,Dest)$, set $M_d(Sour)$, a vertex $last \in M_d(Sour) \cap Dest$ and the array $Prev[VertexNumber]$, where $Prev[N(a)] = 0$ if $a \notin M_d(Sour)/Sour$ and $Prev[N(a)] = b$ otherwise, where (b,a) is the last edge in a least-penalty path connecting the set *Sour* and the vertex a.

30. A method for routing a multi-layered integrated circuit wherein arbitrary routing directions are supported on an arbitrary number of layers of the integrated circuit, comprising:

receiving parameters for an integrated circuit having n layers, wherein n is at least two;

constructing a routing graph for layers of the integrated circuit, the levels partitioned into tiles, wherein at least one edge is provided to join a first tile and a second tile in the routing graph, the tiles positioned generally corresponding to a layer grid line of the level; and

routing based on the routing graph.

31. The method as described in claim 30, wherein routing nets includes routing nets in a net routing area and routing open nets in all integrated circuit areas.

32. The method as described in claim 30, wherein the parameters include at least one of netlist, grid lines slop, distance between neighboring grid lines, and wire blockage information.

33. The method as described in claim 30, wherein a tile is a square section, with one side of each tile of the layer directed corresponding to a layer grid line.

34. The method as described in claim 33, wherein grid lines are positioned generally through a middle of a tile.

35. The method as described in claim 30, wherein the plane of the layer is divided by parallel lines $(y\ cos\alpha_i) + (x\ sin\alpha_i) = j\ d_i$, $j = 0, \pm1, \pm2, \ldots$, where α_i is an inclination of the lines, d_i is the distance between neighboring lines.

36. A system for routing a multi-layered integrated circuit, comprising:

a memory suitable for storing a program of instructions; and

a processor communicatively coupled to the memory, wherein the program of instructions configures the processor to

receive parameters for an integrated circuit having *n* layers, wherein *n* is at least two;

construct a routing graph for layers of the integrated circuit, the levels partitioned into tiles, wherein at least one edge is provided to join a first tile and a second tile in the routing graph, the tiles positioned generally corresponding to a layer grid line of the level; and

route based on the routing graph.

37. The system as described in claim 36, wherein routing nets includes routing nets in a net routing area and routing open nets in all integrated circuit areas.

38. The system as described in claim 36, wherein routing nets includes arriving at a set of edges such that any two different pins of the net are connected by the set.

39. The method as described in claim 36, wherein routing nets includes implementing a procedure to grow neighborhoods.

40. A system for routing a multi-layered integrated circuit, comprising:

a means for receiving parameters for an integrated circuit having *n* layers, wherein *n* is at least two;

a means for constructing a routing graph for layers of the integrated circuit, the levels partitioned into tiles, wherein at least one edge is provided to join a first tile and a second tile in the routing graph, the tiles positioned generally corresponding to a layer grid line of the level; and

a means for routing based on the routing graph.